Updated EAST Search History LTD

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S80	20	(US-5592683-$ or US-5689625-$ or US-5752039-$ or US-5991760-$ or US-6003069-$ or US-6014135-$ or US-6058426-$ or US-6167567-$ or US-6173295-$ or US-6219669-$ or US-6269382-$ or US-6332150-$ or US-6515988-$ or US-6600569-$ or US-6601102-$ or US-6606708-$ or US-6615234-$ or US-6633395-$ or US-6636889-$ or US-6791703-$).did.	USPAT	OR	OFF	2005/06/28 17:29
S81	44	mimeo	USPAT	OR	OFF	2005/06/28 18:50
S56	1	"5592683".pn.	USPAT	OR	OFF	2005/06/28 18:50
S82	53	mimeo	US-PGPUB; USPAT	OR	OFF	2005/06/28 18:54
S83	7	"mimeo.com"	US-PGPUB; USPAT	OR	OFF	2005/06/28 19:41
S88	20758	xml	US-PGPUB; USPAT	OR	OFF	2005/06/28 19:42
S87	309253	request$3	US-PGPUB; USPAT	OR	OFF	2005/06/28 19:42
S86	22422	upload$3	US-PGPUB; USPAT	OR	OFF	2005/06/28 19:42
S85	51725	(active adj server adj page) or asp	US-PGPUB; USPAT	OR	OFF	2005/06/28 19:42
S84	57	ASP with upload$3	US-PGPUB; USPAT	OR	OFF	2005/06/28 19:42
S91	70528	S85 S88	US-PGPUB; USPAT	OR	OFF	2005/06/28 19:43
S90	1347	S86 near4 S87	US-PGPUB; USPAT	OR	OFF	2005/06/28 19:43
S89	139	(S85 S88) near4 S86	US-PGPUB; USPAT	OR	OFF	2005/06/28 19:43
S92	28	S91 same S90	US-PGPUB; USPAT	OR	OFF	2005/06/28 20:02
S96	80355	S85 java	US-PGPUB; USPAT	OR	OFF	2005/06/28 20:26
S95	3679	S86 near2 S94	US-PGPUB; USPAT	OR	OFF	2005/06/28 20:26
S94	1777518	document$1 file$1 job$1	US-PGPUB; USPAT	OR	OFF	2005/06/28 20:26
S93	73	java near4 S86	US-PGPUB; USPAT	OR	OFF	2005/06/28 20:26
S98	13	create with upload with object	US-PGPUB; USPAT	OR	OFF	2005/06/28 20:31

S97	113	S95 same S96	US-PGPUB; USPAT	OR	OFF	2005/06/28 20:31
S101	2294	S96 same S100	US-PGPUB; USPAT	OR	OFF	2005/06/28 20:42
S100	89320	creat$3 with object$1	US-PGPUB; USPAT	OR	OFF	2005/06/28 20:42
S99	85	creat$3 with upload$3 with object$1	US-PGPUB; USPAT	OR	OFF	2005/06/28 20:42
S102	22	S86 same S101	US-PGPUB; USPAT	OR	OFF	2005/06/28 20:43
S104	18	S103 same S86	US-PGPUB; USPAT	OR	OFF	2005/06/28 20:44
S103	2327	GUID	US-PGPUB; USPAT	OR	OFF	2005/06/28 20:44
S105	114	S103 and S86 and S96	US-PGPUB; USPAT	OR	OFF	2005/06/28 20:47
S107	0	".asp"	US-PGPUB; USPAT	OR	OFF	2005/06/28 20:48
S106	2	"upload.begin"	US-PGPUB; USPAT	OR	OFF	2005/06/28 20:48
S108	1	("6458394").PN.	USPAT	OR	OFF	2005/06/29 09:13
S110	4658127	object$1	US-PGPUB; USPAT; USOCR; EPO	OR	OFF	2005/06/29 09:25
S109	1087	http adj post	US-PGPUB; USPAT; USOCR; EPO	OR	OFF	2005/06/29 09:25
S112	22966	upload$3	US-PGPUB; USPAT; USOCR; EPO	OR	OFF	2005/06/29 09:59
S111	123	S109 same S110	US-PGPUB; USPAT; USOCR; EPO	OR	OFF	2005/06/29 09:59
S114	4	S112 same S113	US-PGPUB; USPAT; USOCR; EPO	OR	OFF	2005/06/29 10:05
S116	4	S113 and S114	US-PGPUB; USPAT; USOCR; EPO	OR	OFF	2005/06/29 10:06

S115	2638272	document or print$3	US-PGPUB; USPAT; USOCR; EPO	OR	OFF	2005/06/29 10:06
S113	80	asp near2 object	US-PGPUB; USPAT; USOCR; EPO	OR	OFF	2005/06/29 10:06
S117	57	S113 and S115	US-PGPUB; USPAT; USOCR; EPO	OR	OFF	2005/06/29 10:45
S122	2	S120 and S121	US-PGPUB; USPAT	OR	OFF	2005/06/29 14:48
S121	51615	asp	US-PGPUB; USPAT	OR	OFF	2005/06/29 14:48
S119	64987	session	US-PGPUB; USPAT	OR	OFF	2005/06/29 14:48
S118	22422	upload$3	US-PGPUB; USPAT	OR	OFF	2005/06/29 14:48
S124	3638	709/232.ccls. 709/236.ccls. 709/237.ccls. 709/245.ccls.	US-PGPUB; USPAT	OR	OFF	2005/06/29 14:54
S123	5748	709/227.ccls. 709/228.ccls. 709/229.ccls.	US-PGPUB; USPAT	OR	OFF	2005/06/29 14:54
S120	131	S118 near2 S119	US-PGPUB; USPAT	OR	OFF	2005/06/29 14:54
S128	891725	establish$3	US-PGPUB; USPAT	OR	OFF	2005/06/29 14:55
S127	625	S118 and S125	US-PGPUB; USPAT	OR	OFF	2005/06/29 14:55
S126	226	S121 and S125	US-PGPUB; USPAT	OR	OFF	2005/06/29 14:55
S125	8884	S123 S124	US-PGPUB; USPAT	OR	OFF	2005/06/29 14:55
S132	7	S131 and S125	US-PGPUB; USPAT	OR	OFF	2005/06/29 14:56
S129	888	S118 near3 S118	US-PGPUB; USPAT	OR	OFF	2005/06/29 14:56
S135	60273	S121 S125	US-PGPUB; USPAT	OR	OFF	2005/06/29 14:58
S134	823	S118 near6 S133	US-PGPUB; USPAT	OR	OFF	2005/06/29 14:58
S133	522776	id or identification or guid or identifier	US-PGPUB; USPAT	OR	OFF	2005/06/29 14:58
S131	126	S118 near3 S128	US-PGPUB; USPAT	OR	OFF	2005/06/29 14:58

S141	6	S118 same S140	US-PGPUB; USPAT	OR	OFF	2005/06/29 15:01
S140	2839	S137 same S138 same S139	US-PGPUB; USPAT	OR	OFF	2005/06/29 15:01
S139	104189	packet	US-PGPUB; USPAT	OR	OFF	2005/06/29 15:01
S138	64987	session	US-PGPUB; USPAT	OR	OFF	2005/06/29 15:01
S137	119873	tcp ip	US-PGPUB; USPAT	OR	OFF	2005/06/29 15:01
S136	62	S134 and S135	US-PGPUB; USPAT	OR	OFF	2005/06/29 15:01
S1	0	"709486"	US-PGPUB	OR	OFF	2005/06/29 15:08
S149	51615	asp	US-PGPUB; USPAT	OR	OFF	2005/06/29 15:09
S148	7113	microsoft.as.	US-PGPUB; USPAT	OR	OFF	2005/06/29 15:09
S147	11	S144 and S145	US-PGPUB; USPAT	OR	OFF	2005/06/29 15:09
S146	11	S144 and S145	US-PGPUB	OR	OFF	2005/06/29 15:09
S144	85	S142 and S143	US-PGPUB	OR	OFF	2005/06/29 15:09
S143	19880	asp	US-PGPUB	OR	OFF	2005/06/29 15:09
S142	3158	microsoft.as.	US-PGPUB	OR	OFF	2005/06/29 15:09
S152	11	S151 and S150	US-PGPUB; USPAT	OR	OFF	2005/06/29 15:10
S151	22422	upload$3	US-PGPUB; USPAT	OR	OFF	2005/06/29 15:10
S145	14104	upload$3	US-PGPUB	OR	OFF	2005/06/29 15:10
S150	82	S148 and S149	USPAT	OR	OFF	2005/06/29 15:11
S153	31	"5940812"	US-PGPUB; USPAT; USOCR; EPO	OR	OFF	2005/06/29 15:16
S154	800	"session object"	US-PGPUB; USPAT; USOCR; EPO	OR	ON	2005/07/17 13:50

S157	173	S156 and S154	US-PGPUB; USPAT; USOCR; EPO	OR	ON	2005/07/17 13:51
S155	60955	asp	US-PGPUB; USPAT; USOCR; EPO	OR	ON	2005/07/17 13:51
S159	2326372	asp or "active server page" or iis	US-PGPUB; USPAT; USOCR; EPO	OR	ON	2005/07/17 13:52
S156	61486	asp or "active server page"	US-PGPUB; USPAT; USOCR; EPO	OR	ON	2005/07/17 13:52
S158	25	S156 and S154	USPAT; USOCR; EPO	OR	ON	2005/07/17 13:56
S161	2	S154 same S160	USPAT; USOCR; EPO	OR	ON	2005/07/17 13:57
S160	8923	upload$3	USPAT; USOCR; EPO	OR	ON	2005/07/17 13:57
S162	29	S160 and S154	USPAT; USOCR; EPO	OR	ON	2005/07/17 14:01
S165	160	S164 and S154	US-PGPUB; USPAT; USOCR; EPO	OR	ON	2005/07/17 14:02
S164	19017	709/112.ccls. 709/217.ccls. 709/218.ccls. 709/219.ccls. 709/227.ccls. 709/228.ccls. 709/229.ccls. 709/232.ccls. 709/236.ccls. 709/237.ccls. 709/245.ccls. 709/201.ccls. 709/203.ccls.	US-PGPUB; USPAT; USOCR; EPO	OR	ON	2005/07/17 14:02
S166	73	S164 and S154	USPAT; USOCR; EPO	OR	ON	2005/07/17 14:05
S168	140	S167 and S154	US-PGPUB; USPAT; USOCR; EPO	OR	ON	2005/07/17 14:06
S167	660004	print$3	USPAT; USOCR; EPO	OR	ON	2005/07/17 14:06

S170	102057	S167.ti.	USPAT; USOCR; EPO	OR	ON	2005/07/17 14:08
S169	140	S167 and S154	USPAT; USOCR; EPO	OR	ON	2005/07/17 14:08
S171	3	S170 and S154	USPAT; USOCR; EPO	OR	ON	2005/07/17 14:09
S176	566	S156 same file	USPAT; USOCR; EPO	OR	ON	2005/07/17 14:11
S175	17178	response near3 (id identification track$3 GUID token)	USPAT; USOCR; EPO	OR	ON	2005/07/17 14:11
S174	3944	server near3 (transmit$4 issu$4 send$3 sent) near3 (id identification track$3 GUID token metadata)	US-PGPUB; USPAT; USOCR; EPO	OR	ON	2005/07/17 14:11
S173	3784	server near3 (transmit$4 issu$4 send$3 sent) near3 (id identification track$3 GUID token)	US-PGPUB; USPAT; USOCR; EPO	OR	ON	2005/07/17 14:11
S172	1083	server near3 (transmit$4 issu$4 send$3 sent) near3 (id identification track$3 GUID token)	USPAT; USOCR; EPO	OR	ON	2005/07/17 14:11
S179	44	S174 same S175	USPAT; USOCR; EPO	OR	ON	2005/07/17 14:12
S178	175	S174 and S175	USPAT; USOCR; EPO	OR	ON	2005/07/17 14:12
S177	60	S176 and S160	USPAT; USOCR; EPO	OR	ON	2005/07/17 14:12
S180	181	"HTTP POST" and ASP	US-PGPUB; USPAT; USOCR; EPO	OR	ON	2005/07/21 10:31
S182	95	upload adj complete	US-PGPUB; USPAT; USOCR; EPO	OR	ON	2005/07/21 11:12
S181	38	"HTTP POST" same ASP	US-PGPUB; USPAT; USOCR; EPO	OR	ON	2005/07/21 11:12

S183	30	upload adj complete	USPAT; USOCR; EPO	OR	ON	2005/07/21 11:13
S188	1249	S186 same S187	US-PGPUB; USPAT; USOCR; EPO	OR	ON	2005/07/21 12:12
S187	59432	crc	US-PGPUB; USPAT; USOCR; EPO	OR	ON	2005/07/21 12:12
S186	92686	ack or acknowledge	US-PGPUB; USPAT; USOCR; EPO	OR	ON	2005/07/21 12:12
S185	139283	packet	US-PGPUB; USPAT; USOCR; EPO	OR	ON	2005/07/21 12:12
S184	35538	tcp/ip	US-PGPUB; USPAT; USOCR; EPO	OR	ON	2005/07/21 12:12
S191	33	S184 and S189	USPAT; USOCR; EPO	OR	ON	2005/07/21 12:13
S190	83	S184 and S189	US-PGPUB; USPAT; USOCR; EPO	OR	ON	2005/07/21 12:13
S189	610	S185 same S188	US-PGPUB; USPAT; USOCR; EPO	OR	ON	2005/07/21 12:13

U.S. DEPARTMENT OF COMMERC
Patent and Trademark Offi

DOCUMENT RETRIEVAL REQUEST FORM

Please include RightFax Number to expedite return of documents

Requester's Name: Lucas Divine	Case Serial Number: 09/709,486	Art Unit/Org.:
Phone:	**RightFax:	Building: KxIx	Room Number: 9D 28

Class/Sub-Class:

Date of Request: | Date Needed By:

Paste or add text of citation or bibliography: Paste Citation Only one request per form. Original copy only. ☐

Author/Editor:	
Journal/Book Title:	
Article Title:	
Volume Number:	Report Number: Pages:
Issue Number:	Series Number: Year of Publication:
Publisher:	
Remarks:	Attached

Staff Use Only

Monthly Accession Number:

Library Action	PTO		LC		NAL		NIH		NLM		NIST		Other	
	1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd
Local Attempts														
Date														
Initials														
Results														
Examiner Called														
Page Count														
Money Spent														

	Source	Date
Ordered From:		

Remarks/Comments
1st and 2nd denotes time taken to a library

O/N – Under NLM means Overnight

Comments:

PTO-XXXX (2-96)

USCOM-DC

017020865/9
DIALOG(R)File 350:Derwent WPIX

017020865 **Image available**
WPI Acc No: 2005-345182/200535
Related WPI Acc No: 2002-454234; 2003-670941; 2003-670942; 2003-829596; 2003-829597; 2003-843221; 2005-252893; 2005-260598; 2005-261311; 2005-331721
XRPX Acc No: N05-282089

Web services transfer server for host system used in business application, performs upload method to store binary object, and process binary object method to determine data elements within binary contents of **file for loading data elements**

Patent Assignee: BOTTOMLINE TECHNOLOGIES INC (BOTT-N)
Inventor: CAMPBELL E; HOFFMAN R F; LEMANIS M N; MALONEY R; MINTZER A
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
US 20050097041	A1	20050505	US 200241513	A	20020108	200535 B
			US 2002139596	A	20020506	
			US 2004879406	A	20040629	

Priority Applications (No Type Date): US 2004879406 A 20040629; US 200241513 A 20020108; US 2002139596 A 20020506
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 20050097041	A1		34	G06F-017/60	CIP of application US 200241513 CIP of application US 2002139596

Abstract (Basic): US 20050097041 A1

NOVELTY - The server (46) performs upload method for storing binary object in object storage, associating binary object with object identification (ID) value and returning object ID value to client system (13). A process binary object method is performed for determining data elements within binary contents of file, and loading data elements into application table in accordance with loading rules.

DETAILED DESCRIPTION - An INDEPENDENT CLAIM is also included for method of operating web service transfer server.

USE - For host system to transfer web services to remote transfer client through open network e.g. internet, used in business applications to record commercial interaction with customer, vendor and financial institutions.

ADVANTAGE - The data files are transferred automatically to the client system by loading the data elements in application table. Prevents duplicate downloading of same binary large object (BLOB) by storing offset value assigned to BLOB.

DESCRIPTION OF DRAWING(S) - The figure shows the block diagram of the unattended secure file transfer system.

remote file transfer (10)
host system (11)
internet (12)
client system (13)
web service server (46)
pp; 34 DwgNo 1/17

Title Terms: WEB; SERVICE; TRANSFER; SERVE; HOST; SYSTEM; BUSINESS; APPLY; PERFORMANCE; METHOD; STORAGE; BINARY; OBJECT; PROCESS; BINARY; OBJECT; METHOD; DETERMINE; DATA; ELEMENT; BINARY; CONTENT; FILE; LOAD; DATA; ELEMENT
Derwent Class: T01; W01

International Patent Class (Main): G06F-017/60
File Segment: EPI
Manual Codes (EPI/S-X): T01-J05B4P; T01-N01A1; T01-N01A2A; T01-N01D;
T01-N02A3C; T01-N02B1D; T01-N03A1; W01-A05A; W01-A05B

6357707/9
DIALOG(R)File 2:INSPEC

6357707 INSPEC Abstract Number: C1999-10-7160-037
Title: WebFlow: decentralized workflow management in the World-Wide Web
Author(s): Weber, M.; Illmann, T.; Schmidt, A.
Author Affiliation: Dept. for Distributed Syst., Ulm Univ., Germany
Conference Title: Proceedings of the 16th IASTED International Conference. Applied Informatics p.99-101
Editor(s): Hamza, M.H.
Publisher: IASTED/ACTA Press, Anaheim, CA, USA
Publication Date: 1998 Country of Publication: USA 451 pp.
ISBN: 0 88986 250 8 Material Identity Number: XX-1998-01501
Conference Title: Proceedings of 16th IASTED International Conference on Applied Informatics
Conference Sponsor: IASTED
Conference Date: 23-25 Feb. 1998 Conference Location: Garmisch-Partenkirchen, Germany
Language: English Document Type: Conference Paper (PA)
Treatment: Applications (A); Practical (P)
Abstract: We introduce a workflow management system, called WebFlow, which is based on the World-Wide Web and Java as its basic technologies. Java is used as the build time (modeling) language to define workflows as well as the implementation language for the run time workflow enactment. Due to the object-orientation of Java, modular and extendible workflow types are possible. Modification of workflows is supported even at run time. Using the WWW and Java eases the implementation effort of the workflow engine, since HTTP and the Java API already include functionality which needs not to be implemented anew. This is uploading and downloading of workflow applets and documents, authentication of clients, digital signing and especially the execution of workflows at the client site by the Java virtual machine. Thus, a very simple control server is sufficient, since the applets constituting the workflow coordinate themselves to a large extent. Webflow aims at application scenarios requiring flexible and modifiable workflows. It supports workflows which cross organizational boundaries, since it only relies on standard WWW mechanisms. (3 Refs)
Subfile: C
Descriptors: client-server systems; hypermedia; information resources; Java; workflow management software
Identifiers: WebFlow; decentralized workflow management; World-Wide Web; Java; modeling language; run time workflow enactment; object-orientation; modular workflow types; extendible workflow types; WWW; HTTP; API; applets; client authentication; digital signing; virtual machine; control server; organizational boundaries
Class Codes: C7160 (Manufacturing and industrial administration); C7210N (Information networks); C6110J (Object-oriented programming)

?

WEBFLOW: DECENTRALIZED WORKFLOW MANAGEMENT IN THE WORLD WIDE WEB

MICHAEL WEBER, TORSTEN ILLMANN, ALBRECHT SCHMIDT
Department for Distributed Systems
University of Ulm, Germany

ABSTRACT

In this paper we introduce a workflow management system, called WebFlow, which is based on the world wide web and Java as its basic technologies. Java is used as the build time (modeling) language to define workflows as well as the implementation language for the run time workflow enactment. Due to the object-orientation of Java modular and extendible workflow types are possible. Modification of workflows is supported even at run time. Using WWW and Java eases the implementation effort of the workflow engine, since HTTP and the Java API already include functionality which needs not to be implemented anew. This is uploading and downloading of workflow applets and documents, authentication of clients, digital signing and especially the execution of workflows at the client site by the Java virtual machine. Thus, a very simple control server is sufficient, since the applets constituting the workflow coordinate themselves to a large extent. Webflow aims at application scenarios requiring flexible and modifiable workflows. It supports workflows which cross organizational boundaries, since it only relies on standard WWW mechanisms.

KEYWORDS

workflow management, decentralization, self-coordination

1 INTRODUCTION

Business processes which are optimized in time and flow are becoming crucial for the commercial success of companies. Such processes can be modeled as a composition of activities and subsequently mapped onto workflow management systems [1]. This phase is also called build time. During the flow of business processes the corresponding tasks constitute a so-called workflow which is controlled and coordinated by the workflow management system during run time, the second phase of workflow management. The system allocates persons to activities when the activities become active and schedules them to be performed by the individual persons. To perform activities a person is provided with computer-based resources. A feature of most current workflow management systems is that the entire workflow is defined before it is actually activated. Escaping from the modeled workflow usually is impossible. From an architectural perspective workflow management systems are mostly client/server systems. A centralized workflow engine coordinates the workflow clients being related to the workflow participants.

The growth of the world wide web (WWW) has caused an increasing shift towards the exploitation of its technology for many kinds of applications, also for workflow management systems. Independence of operating system and hardware platform at the client side is guaranteed when using the WWW protocols and document formats. Additionally the paradigm of agent-based distributed systems is evolving. Agents are active instances which are allowed to be executed at a location different from their original host. When using agents in a workflow scenario the necessary control takes place at the client and not through interaction with the central server. The inherent decentralization achieves a far-reaching decoupling of workflow clients and server.

We propose a workflow management system, called WebFlow, entirely based on the WWW and Java [2] as its basic technologies. Java is used as the build time language as well as the implementation language for workflow enactment. Modular and extendible workflow definitions are possible due to the object-orientation of Java. Webflow aims at application scenarios requiring flexible and run time modifiable workflows which may cross organizational boundaries.

2 MODELING OF WORKFLOWS

Workflows are modeled in Java instead of using an extra definition language. Thus workflows are defined in an object-oriented way and maybe easily reused, extended and adapted to re-engineered workflows. No interpreter is required to execute workflow instances. The Java Virtual Machine [3] will do the interpretation.

In WebFlow, abstract workflow classes are provided which already implement the general execution procedure of a workflow. A new workflow is defined by inheriting the abstract class and implementing abstract methods defining the incoming (`initDataflow`) and outgoing data flow (`putDataflow`), the outgoing control flow (`putCon-`

trolflow) and the initialization of an application to be used for the activity (initApplication). The abstract methods of an activity are:

```
abstract class Workflow {
   void initApplication(Application a);
   void initDataflow();
   void putDataflow();
   void putControlflow();
   ... }
```

Workflows are being developed hierarchically. A modeler may specify a single activity or a composite workflow. A composite workflow contains at least two activities: the start and the end activity. After executing the start activity, subworkflows can be defined and optionally activated. When all subworkflows have been finished, the end activity is started in order to evaluate the results of the subworkflows, perform hierarchical data flow and coordinate workflows which are concurrent to this workflow. Defining subworkflows is done within the method defineSubworkflows. For instance, to specify two subworkflows „flow1" and „flow2" the following code is required:

```
void defineSubworkflows(Workflows subflows) {
   subflows.add("flow1","MyWorkflow1.class");
   subflows.add("flow2","MyWorkflow2.class");
}
```

„Flow1" and „flow2" are so-called workflow variables and „MyWorkflow1.class" and „MyWorkflow2.class" are the names of the actual Java classes. This abstraction enables to change a workflow even at run time without modifying every subworkflow referring to it.

One main task of workflow management is the coordination of applications. In WebFlow, an activity refers to an application class providing a set of standard work items. These work items include reporting messages, handling dialogs, accessing databases via JDBC, uploading, downloading and modifying documents, e-mailing, etc. To integrate external applications, the application class has to be inherited and the execute method to be overwritten. E.g.: the start of a local word processing program with a certain document could be implemented as follows:

```
class LocalApplication extends Application {
  Object execute (Object data) {
    String filename = (String) data;
    Runtime rt = Runtime.getRuntime();
    Process p = rt.exec("WordProcessing" +
                        filename);
    return p;
  }
}
```

The execute method is called automatically after starting a workflow and enables data flow between workflow and application. Java enables the access of remote applications through protocols like RMI, RPC, CORBA and JDBC.

To coordinate the control flow of concurrent workflows, three low-level primitives are provided. InsertLink activates a workflow instance for a certain user and inserts a link into his work-to-do-list. RemoveLink finishes a workflow instance and removes the corresponding link from the user's work-to-do-list. UpdateLink updates a workflow instance, i.e. changes user, workflow class, application class, document or time constraint. These primitives enable to define more complex control flow primitives like serialization, parallelism, alternative execution, and joining of parallel workflows.

All concurrent workflows can read and write to a shared database. The workflow class supplies the methods putData and getData to enable this data flow. To perform hierarchical data flow, a composite workflow's start and end activity is allowed to access the database of subworkflows as well as the one of concurrent workflows.

Each composite workflow has to define and assign the workflow participants being available for its subworkflows. Since definition and assignment of workflow participants is done after the execution of the start activity, the actual assignment can be done in a fixed or in a dynamic way.

2.1 MODIFICATION OF WORKFLOWS

The modification of workflows is an important requirement of flexible workflow management systems. Two different aspects may be considered. On the one hand workflow definitions can be modified, on the other hand workflow instances.

Workflow definitions are Java classes which can be modified in two ways. Either the code is changed or it is inherited and extended. The direct change of the source code has the advantage that all active workflow instances are dynamically updated, too. The disadvantage is that the old workflow definition is lost and inconsistencies may be introduced. The recommendable way is to inherit and extend the class. Thus, valuable parts of the old definition can be obtained and new functionality added. The old definition still exists and workflow instances of both definitions can be created. However, active instances are not changed on the fly this way. The following paragraph describes a solution to this problem.

Workflow instances in WebFlow are parameterized Java classes. The instance's data consists of the names of the workflow and the application class, the document URL, the user executing it, and time constraints. The control data consists of the actual state, a unique id, the names of concurrent workflows and the concurrent workflow participants. A workflow currently running subworkflows can

control and influence the execution of these subworkflows. Workflow operations can change their state, i.e. (re-)activate, abort, pause, or resume them. In addition, the accompanying document, the workflow class, or the application may be exchanged on the fly. It is possible to reassign workflow participants. These features are possible, since subworkflows are defined through workflow variables and workflow participants through agents.

3 IMPLEMENTATION ARCHITECTURE

The key idea of WebFlow is the realization of workflows with Java applets. Since Java applets are locally executed programs loaded over the network together with an HTML document, they are an ideal concept for decentralized, self-coordinating workflows. Each applet in WebFlow is able to call local and remote applications and to contact the workflow engine to invoke the subsequent control flow.

The WebFlow system (figure 1) consists of the WebFlow engine, a database management system and a web server on the server machine. The client side comprises a web browser and local applications. Remote applications on third-party computers can be invoked as well.

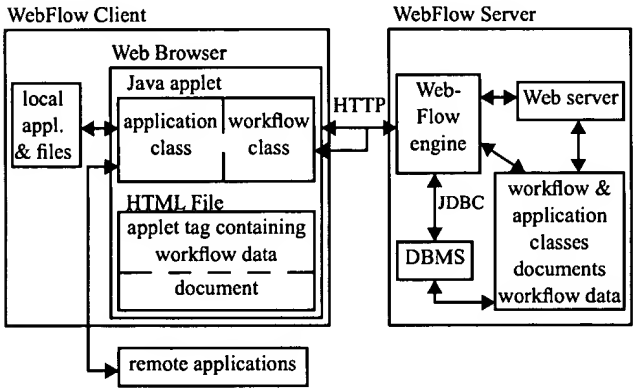

Figure 1: Architecture of the WebFlow system

The execution of a workflow instance proceeds as follows. The user willing to start a workflow loads his work-to-do-list into the web browser. The work-to-do-list is a HTML file containing descriptions of active workflow instances and links (URLs) pointing to the instances themselves which are again HTML documents. Activating a link means to load the document and start the corresponding workflow instance. The web browser does not communicate with the web server directly but through the workflow agent, a part of the WebFlow engine. The workflow agent pipes HTTP requests from the web browser to the web server and HTTP replies the opposite way. Whenever a workflow document is requested, the workflow agent gets the document from the server and attaches the corresponding workflow applet by inserting an HTML APPLET tag to the document stream. Required workflow data is added to the APPLET tag as parameters. The workflow applet is executed automatically on the client machine in the web browser's environment. There, it may perform tasks like executing a dialog with the user, calling local or remote applications, up- or downloading of documents, sending an email or coordinating the execution of concurrent workflow instances belonging to the same superworkflow. With finishing the work items, the workflow applet contacts the workflow engine to perform a transaction of control- and dataflow statements coordinating the concurrent workflows.

One design objective of the WebFlow engine was to place as much intelligence as possible to the decentral, agent-based Java workflows and keep the intervention of the (central) engine as low as possible. Nevertheless, the engine has to do the following tasks: dynamically create work-to-do-lists when requested, authenticate users, transmit workflow instances to the clients, process and reply to workflow requests, control time constraints, and run workflows that should be executed automatically by a non-human participant.

4 CONCLUSIONS

WebFlow is a workflow management system based on the WWW and Java as its basic technologies. Java is used as the build time (modeling) language as well as the implementation language for workflow enactment. In WebFlow modular and extendible workflow definitions are possible due to the object-orientation of Java. Inheritance of already modeled workflow defintions allows an easy construction of sophisticated workflows keeping the modeling process still manageable. Modification of workflows is possible even at run time.

Using WWW and Java eases the implementation effort of the workflow engine, since HTTP and the Java API already include functionality which has not to be implemented anew, such as uploading and downloading of workflow applets and documents, authentication of clients, application of digital signatures and especially the execution of workflows at the client site by the Java virtual machine. I.e. a simple control server is sufficient, since the applets constituting the workflow coordinate themselves to a large extent.

5 REFERENCES

[1] S. Jablonski and C. Bussler, *Workflow Management Modeling Concepts, Architecture and Implementation* (London: International Thomson Computer Press, 1996).

[2] K. Arnold and J. Gosling, *The Java Programming Language* (New York: ACM Press Books, 1996).

[3] T. Lindholm and F. Yellin, *The Java Virtual Machine Specification* (New York: ACM Press Books, 1996).